

TITLE OF THE INVENTION
A PROGRAM EXECUTING MANAGEMENT SYSTEM,
A COMPUTER PROGRAM PRODUCT AND
A PROCESS EXECUTING MANAGEMENT METHOD

5

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is based upon and claims the benefit of priority from the prior Japanese Patent Application No. 2000-168993, June 6, 2000, the entire contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

一般的に、コンピュータシステムは、複数のトランザクション（意味のある処理単位）をシーケンシャルに実行してユーザから要求された処理を実行する。

一般的なデータベース管理システム（DBMS）でも、トランザクションを利用し、制御が実行される。例えば、データベースへのアクセス制御がトランザクションによって実行される。

各種業務を遂行するためのアプリケーション・プログラムを業務アプリケーションとする。例えば、業務アプリケーションの例として、ホテルと航空券の予約状態を検索し、予約を行う処理がある。このような各種の業務アプリケーションにおいてもトランザクション制御を利用している。

コンピュータシステムにおけるトランザクション制御においては、以下のACID（Atomicity, Consistency, Isolation, Durability）特性を保証することが重要である。

Atomicityは、そのトランザクションが実行されるか、まったく実行されないかのいずれかを確保する特性をいう。

Consistencyは、トランザクションが終了したときに、データベースの内容を矛盾のない状態とする特性をいう。

Isolationは、トランザクション実行途中の内容が他のトランザクションから

観測されない特性をいう。

Durabilityは、実行が終了したトランザクションの結果が消えない特性をいう。

このように、従来のコンピュータシステムにおいては、ACID特性を保証し、トランザクションの単位で正常な処理の実現を図っている。

5 その一方で、ユーザがコンピュータシステムに対して実行を要求する対象は、業務を遂行するための処理、すなわち業務アプリケーションである。

したがって、ユーザは、業務を遂行する上でこの業務に必要な処理が全て実行されたか否か等の保証を望む傾向がある。

しかしながら、システムが提供するトランザクションの単位と業務単位には大きな隔りがある。また、保証すべき事項も業務に応じて異なる。このため、業務アプリケーションの処理が複雑化し、業務アプリケーションの開発に労力がかかる。

例えば、上記のホテルと航空券を予約するための業務アプリケーションでは、この業務遂行の途中で人間による入力が必要な場合がある。すなわち、予約状況の表示を見て予約内容を決定制し、入力を行うのは人間である。

このように業務遂行途中で人間の判断が入る場合、一般的に、コンピュータシステム側ではその処理を人間からの入力前と入力後で別々のフローとして扱う。

この場合において人間が適切な指示をコンピュータシステムに与えなかったとすると、人間の入力前までの処理は実行されているがその後の処理が実行されず、
20 業務が途中で中断されることになる。

従来のコンピュータシステムにおいては、このような業務の中断を検出することが困難である。そして、このような業務の中断を防止するために、業務アプリケーションに各種の例外処理及びエラーチェックを挿入する必要がある。

しかしながら、業務アプリケーションにこのような例外処理及びエラーチェックを含めると、業務アプリケーションの複雑化を招く。
25

BRIEF SUMMARY OF THE INVENTION

本発明は、処理の簡略化を図りつつ処理の正常性を保証するプログラム実行管理システム、プログラムプロダクト、及びコンピュータシステムによる処理の実

行管理方法を提供することを目的とする。

第1の発明は、シーケンシャルに複数の処理要素を利用する処理についての識別情報毎に、その処理で利用される複数の処理要素の識別情報とその実行順序を記憶する処理要素定義ユニットと、複数の処理要素の識別情報毎に、異常が発生した場合に実行すべき復元処理要素の識別情報を記憶する復元処理要素定義ユニットと、処理からの処理要素利用要求に基づいて、処理要素定義ユニットを参照して次に処理すべき処理要素の識別情報を求め、求めた識別情報に対応する処理要素を実行する実行ユニットと、実行ユニットによる処理要素の実行において異常が発生した場合に、この異常の発生した処理要素の識別情報に基づいて復元処理要素定義ユニットを参照し、この異常に対応する復元処理要素を実行する復元実行ユニットとを具備する。

この第1の発明においては、処理要素がシーケンシャルに利用されて業務が遂行される。また、第1の発明では、複数の処理要素毎に復元処理要素が定められており、異常が発生した場合にこの異常の発生した処理要素に対応する復元処理要素が実行される。

これにより、業務アプリケーション等の処理自体に複雑な例外処理とエラーチェックを含めなくても処理の正常性が確保でき、プログラムの開発労力を軽減できる。

第2の発明では、第1の発明における処理をアプリケーションが実行する。

第3の発明は、第1の発明に、処理の実行要求が発生した場合にこの実行要求に対する固有番号を発番する発番ユニットを具備する。そして、実行ユニットは、発番ユニットによって発番された固有番号毎に処理からの処理要素利用要求を区別し、この固有番号毎に実行済みの処理要素の数を管理し、この実行済みの処理要素の数に基づいて処理要素定義ユニットを参照し、次に実行すべき処理要素の識別情報を求め、求めた識別情報に対応する処理要素を実行する。

この第3の発明においては、処理の実行要求毎に固有番号が付され、この実行要求毎に処理要素の実行順序が管理されるため、同じ処理の実行が複数回要求されても個別に処理を管理でき、異常が発生しても個別に復元できる。

第4の発明は、シーケンシャルに複数の処理要素を利用する処理についての識

別情報毎に、その処理で利用される複数の処理要素の識別情報とその実行順序を記憶する処理要素定義ユニットと、複数の処理要素の識別情報毎に、異常が発生した場合に実行すべき復元処理要素の識別情報を記憶する復元処理要素定義ユニットと、処理の実行要求が発生した場合にこの処理の識別情報に基づいて処理要素定義ユニットを参照し、この処理の識別情報に対応する複数の処理要素をその実行順序にしたがって順次実行する実行ユニットと、実行ユニットによる処理要素の実行において異常が発生した場合に、この異常の発生した処理要素の識別情報に基づいて復元処理要素定義ユニットを参照し、この異常に対応する復元処理要素を実行する復元実行ユニットとを具備する。

この第4の発明においては、1回の処理の実行要求に対して複数の処理要素が実行される。

第5から第8までの発明は、上記第1から第4までの発明の各ユニットと同様の機能を実現するプログラムコードを具備したコンピュータプログラムプロダクトである。

このコンピュータプログラムプロダクトを用いることによって、上述した機能を備えていないコンピュータシステム、サーバなどに対して簡単に上記した機能を付加することができる。

第9の発明は、コンピュータ読み取り可能なコンピュータプログラムプロダクトである。この第9の発明のプログラムプロダクトは、シーケンシャルに複数の処理要素を利用する処理からの処理利用要求に基づいて、処理についての識別情報毎に記憶されたその処理で利用される複数の処理要素の識別情報とその実行順序を参照して次に処理すべき処理要素の識別情報を求め、求めた識別情報に対応する処理要素を実行する実行コードと、実行コードによる処理要素の実行結果を処理に通知する結果通知コードとを具備する。

この第9の発明を利用することで、処理のうちの一部分を処理要素によって実行し、処理に結果を返すことができる。したがって、この処理についてのプログラムの開発労力を軽減できる。

第10の発明のプログラムプロダクトは、複数の再利用可能な処理要素をシーケンシャルに実行する処理の識別情報と、その処理で利用される複数の処理要素

の識別情報と、複数の処理要素の実行順序とを入力する入力コードと、処理の識別情報毎に、その処理で利用される複数の処理要素の識別情報とその処理で利用される複数の処理要素の実行順序とを記憶する定義コードとを具備する。

第 11 の発明は、第 10 の発明に、実行すべき処理の識別情報を入力した場合に、定義コードによって記憶された内容を参照し、入力した処理の識別情報に対応する実行順序にしたがってこの入力した処理の識別情報に対応する処理要素を実行する実行コードと、実行コードによる処理要素の実行結果を処理に通知する結果通知コードとを具備する。

この第 10 及び第 11 の発明を利用すれば、再利用可能な処理要素のうち、顧客の利用したい処理要素をシーケンシャルに実行するサービスを提供できる。したがって、顧客は業務アプリケーションを容易に開発できる。

第 12 から第 14 までの発明は、それぞれ上記第 9 から第 11 までの発明におけるプログラムを読み込んだコンピュータシステムによる処理の実行管理方法である。

Additional objects and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The objects and advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out hereinafter.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate presently preferred embodiments of the invention, and together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention.

図 1 は、本発明の第 1 の実施の形態に係るプログラム実行管理システムの構成例を示すブロック図。

図 2 は、第 1 の実施の形態に係るプログラム実行管理システムに備えられている入出力ユニットの動作例を示すフローチャート。

図 3 は、第 1 の実施の形態に係るプログラム実行管理システムに備えられている実行制御ユニットの動作例を示すフローチャート。

図 4 は、第 1 の実施の形態に係るプログラム実行管理システムに備えられている発番ユニットの動作例を示すフローチャート。

図 5 は、本発明の第 2 の実施の形態に係るプログラム実行管理システムに備えられている入出力ユニットの動作例を示すフローチャート。

図 6 は、第 2 の実施の形態に係るプログラム実行管理システムに備えられている実行制御ユニットの動作例を示すフローチャート。

図 7 は、プログラム実行管理システムと同様の動作を実現するためのコンピュータプログラムプロダクトに具備されているプログラムコードを例示するブロック図。

図 8 は、プログラム実行管理システムによるサービスの提供状態を例示するブロック図。

図 9 は、プログラム実行管理システムによって提供される情報処理サービスを例示するフローチャート。

DETAILED DESCRIPTION OF THE INVENTION

以下、図面を参照しながら本発明の実施の形態について説明する。

(第 1 の実施の形態)

本実施の形態においては、業務アプリケーションが複数の処理要素をシーケンシャルに利用し、この処理要素の実行要求をシーケンシャルに発生させ、業務を遂行するプログラム実行管理システムについて説明する。

図 1 は、本実施の形態に係るプログラム実行管理システムの構成例を示すブロック図である。

このプログラム実行管理システム 1 は、シーケンシャルに利用される複数のシーケンス処理要素 2 a ~ 2 g を備え、またこのシーケンス処理要素 2 a ~ 2 g を利用しつつ業務を遂行する業務アプリケーション 3、4 を備える。

業務アプリケーション3、4は、実行が要求されて最初にシーケンス処理要素
を利用する場合に、自己の種別及びシーケンス処理要素へのデータを出力し、実
行要求に対する固有番号とシーケンス処理要素の実行結果を入力する。その後、
シーケンス処理要素を利用する業務アプリケーション3、4は、データと先にお
5 いて入力した固有番号を出力し、シーケンス処理要素の実行結果を入力する。

この図1において、業務アプリケーション3は「ホテル・航空券予約」アプリ
ケーションである。業務アプリケーション4は「レストラン予約」アプリケー
ションである。

また、シーケンス処理要素2 aの「hotel-search」はホテルの予約状況を検索
して表示する処理である。シーケンス処理要素2 bの「air-search」は航空券の
予約状況を検索して表示する処理である。シーケンス処理要素2 cの「hotel-re
serve」はホテルを予約する処理である。シーケンス処理要素2 dの「air-reser
ve」は航空券を予約する処理である。シーケンス処理要素2 eの「rest-a-search
h」はレストランA店の予約状況を検索して表示する処理である。シーケンス処
理要素2 fの「rest-b-search」はレストランB店の予約状況を検索して表示す
る処理である。シーケンス処理要素2 gの「rest-reserve」はレストランを予約
する処理である。

さらに、プログラム実行管理システム1は、シーケンス処理要素のいずれかが
異常終了した場合にこの異常からの復元処理を実行する復元処理要素5 a～5 c
20 を備えている。

図1において、復元処理要素5 aの「hotel-cancel」はホテルの予約をキャン
セルする処理である。復元処理要素5 bの「air-cancel」は航空券の予約をキャン
セルする処理である。復元処理要素5 cの「rest-cancel」はレストランの予
約をキャンセルする処理である。

さらに、プログラム実行管理システム1は、業務アプリケーションからの入力
に応じて該当するシーケンス処理要素の実行を管理し、また異常が発生した場合
に該当する復元処理要素の実行を管理するための発番ユニット6、シーケンス登
録ユニット7、定義記憶ユニット8、入出力ユニット9、実行制御ユニット10、
処理実施ユニット11を備える。

発番ユニット6は、業務アプリケーションの実行要求が発生するたびに、この実行要求に対する固有番号を発番する。

シーケンス登録ユニット7にはシーケンス登録テーブル7aが記憶される。シーケンス登録テーブル7aは、実行要求に対する固有番号毎に、この実行要求された業務アプリケーションの種別とシーケンス処理要素がどこまで実行済みかを示す最終ステップ番号とステータスとを管理する。

定義記憶ユニット8には処理要素定義テーブル8aと復元処理要素定義テーブル8bとが記憶される。

処理要素定義テーブル8aは、業務アプリケーションの種別毎に、利用されるシーケンス処理要素数であるステップ数と実行順序に基づいてステップ番号を付したシーケンス処理要素の種別とを管理する。

復元処理要素定義テーブル8bは、業務アプリケーションの種別毎に、異常の発生したステップ番号における復元処理要素の種別とその数であるステップ数を管理し、さらに復元処理要素の種別に実行順序を示すステップ番号を付して管理する。

入出力ユニット9は、業務アプリケーションがシーケンス処理要素を利用する度に、この業務アプリケーションの種別又は固有番号を入力し、その内容を実行制御ユニット10に出力する。また、実行処理ユニット10からシーケンス処理要素の実行結果と固有番号を入力すると、この内容を該当する業務アプリケーションに出力する。

実行制御ユニット10は、新規に実行要求された業務アプリケーションから入出力ユニット9を介してこの業務アプリケーションの種別を入力すると、この業務アプリケーションの種別に基づき処理要素定義テーブル8aを参照する。そして、実行制御ユニット10は、実行要求に対する固有番号を発番ユニット6に発番させてシーケンス登録テーブル7aに登録し、さらに最初のシーケンス処理要素の実行を処理実施ユニット11に命令し、実行結果及び固有番号を入出力ユニット9に出力する。

一方、実行制御ユニット10は、新規に実行要求されたのではない業務アプリケーションから入出力ユニット9を介して固有番号を入力すると、この固有番号

に基づいて処理要素定義テーブル 8 a の参照とシーケンス登録テーブル 7 a の参照・更新を行い、次のシーケンス処理要素の実行を処理実施ユニット 1 1 に命令し、実行結果を出力ユニット 9 に出力する。

また、この実行制御ユニット 1 0 は、シーケンス処理要素の実行結果を監視し、その結果をシーケンス登録テーブル 7 a のステータスに登録し、実行結果に異常が発生している場合には、復元処理要素定義テーブル 8 b を参照し、実行すべき復元処理要素の実行を処理実施ユニット 1 1 に命令する。

処理実施ユニット 1 1 は、実行制御ユニット 1 0 からの実行命令の示すシーケンス処理要素を実行し、実行結果を実行制御ユニット 1 0 に出力する実行機能 1 1 a を備える。

また、処理実施ユニット 1 1 は、実行命令の示す復元処理要素を実行し、実行結果を実行制御ユニット 1 0 に出力する復元実行機能 1 1 b を備える。

表 1 は、先に述べた処理要素定義テーブル 8 a の内容を例示している。

表 1. 処理要素定義テーブル

業務アプリケーションの種別	ステップ数	シーケンス処理要素の種別
ホテル・航空券予約	4	1.hotel-search 2.air-search 3.hotel-reserve 4.air-reserve
レストラン予約	3	1.rest-a-search 2.rest-b-search 3.rest-reserve

例えば、業務アプリケーション 3 の「ホテル・航空券予約」では、シーケンス処理要素 2 a の「hotel-search」、シーケンス処理要素 2 b の「air-search」、シーケンス処理要素 2 c の「hotel-reserve」、シーケンス処理要素 2 d の「air-reserve」がこの順序で利用される。したがって、表 1 では、「ホテル・航空券予約」におけるシーケンス処理要素のステップ数は 4 である。また、各シーケ

10
05873251
15
060501

表 2 は、先に述べた復元処理要素定義テーブル 8 b の内容を例示している。

表 2. 復元処理要素定義テーブル

業務アプリケーションの種別	異常の発生したステップ番号	実行する復元処理要素のステップ数	復元処理要素の種別
ホテル・航空券予約	1	0	
"	2	0	
"	3	1	hotel-cancel
"	4	2	1.hotel-cancel 2.air-cancel
レストラン予約	1	0	
"	2	0	
"	3	1	1.rest-cancel

例えば、業務アプリケーション3の「ホテル・航空券予約」では、ステップ番号4の付されているシーケンス処理要素2dが異常終了した場合、復元処理要素5aの「hotel-cancel」、復元処理要素5bの「air-cancel」がこの順序で利用される。したがって、表2では、「ホテル・航空券予約」のステップ番号4における復元処理要素のステップ数は2であり、各復元処理要素5a、5bの種別に対してそれぞれステップ番号1、2が付されている。

表3は、先に述べたシーケンス登録テーブル7aの内容を例示している。

表 3. シーケンス登録テーブル

実行要求の 固有番号	業務アプリケーション の種別	実行済みの最終 ステップ番号	ステータス
0001	ホテル・航空券予約	3 (ホテル予約)	
0002	レストラン予約	4	終了

この表 3 では、例えば、固有番号「0001」の実行要求は業務アプリケーション 3 の「ホテル・航空券予約」に対する実行要求である。また、この実行要求についてステップ番号 3 までが実行済みである旨が、その時点でのステータスと共に格納されている。

なお、新たな「ホテル・航空券予約」の実行要求が発生すると、先に発生した「ホテル・航空券予約」とは異なる固有番号により同様に管理される。

上記のような構成を備えるプログラム実行管理システム 1 の動作例について以下に説明する。

業務アプリケーション 3、4 は、シーケンス処理要素の利用を要求する度に、入出力ユニット 9 に対して業務アプリケーションの種別又は固有番号と、必要であればデータを出力し、固有番号と実行結果などをその応答として入力する。

具体的には、最初にシーケンス処理要素の利用を要求する場合、業務アプリケーション 3、4 は、自己の種別と必要なデータを入出力ユニット 9 に出力し、この応答として固有番号と実行結果を入出力ユニット 9 から入力する。

一方、シーケンス処理要素の利用が最初ではない場合、業務アプリケーション 3、4 は、固有番号と必要なデータを入出力ユニット 9 に出力し、この応答として実行結果を入出力ユニット 9 から入力する。

図 2 は、本実施の形態に係るプログラム実行管理システム 1 に備えられている入出力ユニット 9 の動作例を示すフローチャートである。

入出力ユニット 9 は、業務アプリケーションから入力した業務アプリケーションの種別又は固有番号と、必要なデータを実行制御ユニット 10 に出力する（s 1）。

その後、実行制御ユニット 10 の動作が終了すると、入出力ユニット 9 は、実行制御ユニット 10 から固有番号と実行結果などを応答として入力し、この応答をシーケンス処理要素の利用を要求した業務アプリケーションに出力する（s 2）。

具体的には、最初のシーケンス処理要素の利用であった場合、入出力ユニット 9 は、実行結果と固有番号とをこの実行制御ユニット 10 から入力し、シーケンス処理要素の利用を要求した業務アプリケーションに出力する。一方、シーケンス処理要素の利用が最初でなかった場合、入出力ユニット 9 は、実行制御ユニット 10 の動作終了後、この実行制御ユニット 10 から実行結果を入力し、シーケンス処理要素の利用を要求した業務アプリケーションに出力する。

図 3 は、本実施の形態に係るプログラム実行管理システム 1 に備えられている実行制御ユニット 10 の動作例を示すフローチャートである。

実行制御ユニット 10 は、入出力ユニット 9 からの入力内容に固有番号が含まれているか否かを判定する（t 1）。

ここで、入力ユニット 9 からの入力内容に固有番号が含まれていない場合（固

有番号が0の場合)には、実行制御ユニット10は、発番ユニット6に固有番号の発番を要求し(t2)、この発番ユニット6によって固有番号が発番されると、その発番された固有番号をシーケンス登録テーブル7aに新規登録し(t3)、実行すべきシーケンス処理要素のステップ番号を示すproc_numを0とする(t4)。

一方、入力ユニット9からの入力内容に固有番号が含まれている場合には、実行制御ユニット10は、この固有番号に基づいてシーケンス登録テーブル7aを参照し、該当する最終ステップ番号を取得し、proc_numの値とする(t5)。

すると、実行制御ユニット10は、固有番号とproc_num+1の値に基づいて処理要素定義テーブル8aを参照し、この固有番号においてproc_numの値と同じステップ番号のシーケンス処理要素を認識する。また、実行制御ユニット10は、入出力ユニット9からの入力内容に含まれるデータを利用してこの認識したシーケンス処理要素の実行を処理実施ユニット11に要求する(t6)。

そして、処理実施ユニット11によって該当するシーケンス処理要素が実行されると、実行制御ユニット10は、その実行結果が正常か否かを判定する(t7)。

実行結果が正常な場合、実行制御ユニット10は、固有番号に基づいてシーケンス登録テーブル7aを参照し、終了であればステータスにその旨を登録する(t8)。

一方、実行結果が異常な場合、実行制御ユニット10は、固有番号とproc_numの値とに基づいて復元処理要素定義テーブル8bを参照し、この固有番号においてproc_num+1の値と同じステップ番号に対する復元処理要素を認識する。また、実行制御ユニット10は、この認識した復元処理要素の実行を処理実施ユニット11に要求し(t9)、これにより処理実施ユニット11によって該当する復元処理要素が実行される。

そして、実行制御ユニット10は、実行結果を入出力ユニット9に出力する(t10)。

図4は、本実施の形態に係るプログラム実行管理システム1に備えられている発番ユニット6の動作例を示すフローチャートである。

発番ユニット6は、実行制御ユニット10から固有番号の発番を要求されると、シーケンス登録テーブル7aを参照し、このシーケンス登録テーブル7aに登録されている固有番号の最大値に1を加えた値を新規の固有番号として実行制御ユニット10に出力する(u1)。

5 以上説明したように、本実施の形態に係るプログラム実行管理システム1においては、業務アプリケーション3、4がシーケンス処理要素を順次利用しつつ業務を遂行し、異常終了した場合には実行済みのシーケンス処理要素に対応した復元処理要素により復元が実行される。

10 このように、シーケンシャルに処理要素を実行し、異常発生時点までの復元処理を別途用意し実行することにより、異常に対処する例外処理及びエラーチェックのための処理を開発時に業務アプリケーションに含める必要がない。

したがって、業務アプリケーションの複雑化を防止し、開発労力を軽減しつつ、業務遂行に関するACID特性が保証でき、業務処理の正常性を確保できる。

(第2の実施の形態)

15 上記第1の実施の形態においては、業務アプリケーションがシーケンス処理要素の利用を要求する度に入出力ユニットに対して出力を行うとしている。

これに対し、本実施の形態においては、業務アプリケーションからの要求にしたがって、複数のシーケンス処理要素が一括実行されるプログラム実行管理システムについて説明する。

20 このように、業務アプリケーションからの要求に対して複数のシーケンス処理要素を一括実行するプログラム実行管理システムに関しては、上記図1の構成要素のうち、業務アプリケーション、複数のシーケンス処理要素及び復元処理要素、定義記憶ユニット、入出力ユニット、実行制御ユニット、処理実施ユニットが必要であり、このうち主に業務アプリケーション、入出力ユニット、及び実行制御ユニットの動作を変更すればよい。

そこで、本実施の形態に係るプログラム実行管理システムに備えられている業務アプリケーション、入出力ユニット、及び実行制御ユニットの動作について以下に説明する。

業務アプリケーションは、シーケンス処理要素を利用する場合に、入出力ユニ

ットに対して業務アプリケーションの種別と、必要であればデータを出力し、実行結果をその応答として入力する。

図5は、本実施の形態に係るプログラム実行管理システムに備えられている入出力ユニットの動作例を示すフローチャートである。

- 5 入出力ユニットは、業務アプリケーションから入力した業務アプリケーションの種別と必要なデータを実行制御ユニットに出力する。同時に、業務アプリケーションへの結果の返答先をrepに記録する（v1）。

その後、実行制御ユニットの動作が終了すると、入出力ユニットは、実行制御ユニットから実行結果を応答として入力し、この応答をシーケンス処理要素の利用を要求した業務アプリケーションrepに出力する（v2）。

図6は、本実施の形態に係るプログラム実行管理システムに備えられている実行制御ユニットの動作例を示すフローチャートである。

実行制御ユニットは、入力ユニットから入力された業務アプリケーションの種別に基づいてシーケンス登録テーブル7aを参照し、この業務アプリケーションの種別に対応付けされているステップ数を取得し、実行すべきシーケンス処理要素の総数を示すproc_numに格納する（w1）。

また、実行制御ユニットは、カウンタ変数であるcountに1を格納する（w2）。

ここで、countの値がproc_numの値以下か否かを実行制御ユニットは判定する（w3）。

Countの値がproc_numの値以下の場合、実行制御ユニットは、入力した業務アプリケーションの種別に基づいて処理要素定義テーブル8aを参照し、この業務アプリケーションの種別に対応づけされているcount番目のシーケンス処理要素を認識する。また、実行制御ユニットは、入出力ユニットから入力したデータに基づいて認識したシーケンス処理要素の実行を処理実施ユニットに要求する（w4）。

そして、処理実施ユニットによって該当するシーケンス処理要素が実行されると、実行制御ユニットは、その実行結果が正常か否かを判定する（w5）。

実行結果が正常な場合、実行制御ユニットはcountの値に1を加え（w6）、

再びcountの値がproc_numの値以下か否かを判定する（w 3）。

一方、実行結果が異常な場合、実行制御ユニットは、業務アプリケーションの種別とcountの値とに基づいて復元処理要素定義テーブルを参照し、該当する業務アプリケーションにおけるcountの値と同じステップ番号に対する復元処理要素を認識する。また、実行制御ユニットは、この認識した復元処理要素の実行を処理実施ユニットに要求する（w 7）。

これにより処理実施ユニットによって該当する復元処理要素が実行され、実行制御ユニットは「失敗」である旨の実行結果を入出力ユニットに出力する（w 8）。

これに対し、countの値がproc_numの値以下でないと実行制御ユニットが判定した場合には、実行制御ユニットは「完了」である旨の実行結果を入出力ユニットに出力する（w 9）。

以上説明したように、本実施の形態に係るプログラム実行管理システムにおいては、業務アプリケーションが自己の種別を入出力ユニットに出力するのみで、複数のシーケンス処理要素が順次実行される。この複数のシーケンス処理要素の実行で異常が発生しても、その時点に応じた復元処理要素が実行される。

したがって、本実施の形態においては、上記第1の実施の形態と同様に、業務アプリケーションの複雑化を防止し、開発時の労力を軽減しつつ、業務単位で処理の正常性を保証することができる。

なお、上記各実施の形態に係るプログラム実行管理システムは、同様の作用・機能を実現可能であれば各構成要素の配置を変更させてもよく、また各構成要素を自由に組み合わせてもよく、各構成要素をさらに細分化してもよい。

また、本実施の形態に係るプログラム実行管理システムの各ユニット、各要素、各機能は、コンピュータに実行させることのできるプログラムとして、例えば磁気ディスク（フレキシブルディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体に書き込んでコンピュータシステムに適用してもよい。また、このようなプログラムを通信媒体により伝送して、コンピュータシステム、サーバに適用してもよい。

上記各機能を実現するコンピュータは、記録媒体に記録されたプログラムを読

み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

図7は、上記各実施の形態に係るプログラム実行管理システムと同様の動作を実現するためのコンピュータプログラムプロダクト12に具備されているプログラムコードを例示している。各プログラムコード12a～12eは、コンピュータシステム13に読み込まれ、実行されると、以下のように動作する。

処理要素定義コード12aは、シーケンス処理要素を利用するアプリケーションの識別情報毎に、そのアプリケーションにより利用されるシーケンス処理要素の識別情報とその実行順序とを記憶する。

復元処理要素定義コード12bは、シーケンス処理要素の識別情報毎に、異常が発生した場合に実行すべき復元処理要素の識別情報を記憶する。

発番コード12cは、アプリケーションの実行要求単位に、固有番号を発番する。

実行コード12dは、発番コード12cによって発番された固有番号を利用してアプリケーションの動作を実行要求単位で識別する。そして、実行コード12dは、処理要素定義コード12aによって記憶された内容を参照して次に処理すべきシーケンス処理要素の識別情報を求め、求めた識別情報に対応するシーケンス処理要素を実行する。

復元実行コード12eは、実行コード12dによるシーケンス処理要素の実行において異常が発生した場合に、この異常の発生したシーケンス処理要素の識別情報に基づいて復元処理要素定義コード12bによって記憶された内容を参照し、この異常に対応する復元処理要素を実行する。

(第3の実施の形態)

本実施の形態においては、上述したプログラム実行管理システムによって提供される情報処理サービスについて説明する。

図8は、プログラム実行管理システムによるサービスの提供状態を例示するブロック図である。

ユーザ14の操作するクライアント15とASP(Application Service Provider)の管理するサーバ16とプログラム実行管理システム17は、例えばイン

ターネットなどのようなネットワーク 18 を介して互いに送受信可能に接続されている。また、業務アプリケーション 16 a としては、Web Services におけるサービスなどが考えられる。

プログラム実行管理システム 17 は、再利用可能なシーケンス処理要素 19 1 ~ 19 n を予め保持している。プログラム実行管理システム 17 は、例えば課金管理を行うプログラム、ユーザ認証を行うプログラムなどの再利用可能なシーケンス処理要素を保持する。

プログラム実行管理システム 17 は、入力機能 20、定義機能 21、実行機能 22、結果通知機能 24 を持つ。この入力機能 20、定義機能 21、実行機能 22、結果通知機能 24 は、コンピュータプログラムプロダクト 23 から読み込んだ入力コード 23 a、定義コード 23 b、実行コード 23 c、結果通知コード 23 d を実行して実現してもよい。

入力機能 20 は、シーケンス処理要素の代理実行を依頼する者からネットワーク 18 を介して、実際にシーケンス処理要素を代理実行する際に利用される情報を予め入力する。

シーケンス処理要素の代理実行に必要な情報には、例えば処理の識別情報（アプリケーションの識別情報）、その処理で利用されるシーケンス処理要素の識別情報、その処理で利用されるシーケンス処理要素の実行順序などがある。

定義機能 21 は、実際にシーケンス処理要素を代理実行する際に利用される情報を定義記憶ユニット 8 の処理要素定義テーブル 8 a に予め記憶する。

実行機能 22 は、シーケンス処理要素の代理実行を依頼する者からネットワーク 18 を介して、処理の識別情報を含むシーケンス処理要素の実行命令を入力した場合に、処理要素定義テーブル 8 a を参照し、入力した処理の識別情報に対応する実行順序にしたがってこの入力した処理の識別情報に対応するシーケンス処理要素を実行する。

結果通知機能 24 は、シーケンス処理要素の実行結果を業務アプリケーション 16 a に返す。

図 9 は、プログラム実行管理システム 17 によって提供される情報処理サービスを例示するフローチャートである。

まず、ASPは、業務アプリケーション16aに所定の機能を付加することを目的として、再利用可能なシーケンス処理要素191～19nの中から自己の利用したいシーケンス処理要素を選択する。

ASPは、この業務アプリケーション16aの識別情報と、選択したシーケンス処理要素の識別情報と、その実行順序を、プログラム実行管理システム17に送信する。

プログラム実行管理システム17は、入力機能20によってASPからの情報を入力し(x1)、定義機能21によってその情報を処理要素定義テーブル8aに記憶する(x2)。

そして、ユーザ14は自己のクライアント15を操作し、ASPの業務アプリケーション16aを利用する。

ユーザによる業務アプリケーション16aの利用によりシーケンス処理要素の実行依頼イベントが発生すると、シーケンス処理要素の実行命令がプログラム実行管理システム17に送信される。

プログラム実行管理システム17は、シーケンス処理要素の実行命令を入力し(x3)、実行するように依頼されたシーケンス処理要素を実行機能22によって実行し(x4)、結果通知機能24によって結果を返す(x5)。そして、業務アプリケーションは、これらの結果を利用して所定の目的にそって定義された処理を行う。

例えば、業務アプリケーションは、「課金管理」という処理要素から「支払い状態」の情報を受け、「ユーザ承認」という処理要素から「アクセス権限」の情報を受け取る。

そして、この結果、「課金管理」から「支払いが行われている」という情報を受け付け、かつ「ユーザ承認」から「適切なアクセス権限がある」という情報を受け付けた場合、業務アプリケーションは次の処理を実行する。

一方、「課金管理」から「支払いが行われていない」という情報を受け付けた場合、又は「ユーザ承認」から「アクセス権限なし」という情報を受け付けた場合、業務アプリケーションは次の処理を実行しない。

このように、処理要素を利用して新たな処理を定義・実行することが可能とな

る。

以上説明したように、本実施の形態に係るプログラム実行管理システム 17 を利用した場合には、再利用可能なシーケンス処理要素 191 ~ 19n を自由に選択して利用可能である。

したがって、シーケンス処理要素の代理実行を依頼する者は、アプリケーションの複雑化を防止し、開発労力を軽減でき、アプリケーションの機能を充実させることができる。また、シーケンス処理要素の代理実行を依頼する者は、自己でシーケンス処理要素を利用する場合よりも保守、運用の面で効率化を図ることができる。

一方、プログラム実行管理システム 17 の運用者は、シーケンス処理要素の代理実行によりシーケンス処理要素の代理実行を依頼する者から対価を得ることができる。

Additional advantages and modifications will readily occur to those skilled in the art. Therefore, the invention in its broader aspects is not limited to the specific details and representative embodiments shown and described herein. Accordingly, various modifications may be made without departing from the spirit or scope of the general inventive concept as defined by the appended claims and their equivalents.